AD-784 231

NATURAL COMMUNICATION WITH COMPUTERS IV

Bolt Beranek and Newman, Incorporated

Prepared for:

Advanced Research Projects Agency

August 1974

Best Available Copy

B O L T   B E R A N E K   A N D   N E W M A N   I N C

C O N S U L T I N G · D E V E L O P M E N T · R E S E A R C H

BBN Report 2869

July 1974

NATURAL COMMUNICATION WITH COMPUTERS IV

Quarterly Progress Report No. 15

1 April 1974 to 30 June 1974

This research was supported by the Advanced Research Projects Agency under ARPA Order no 1967; Contract No. DAHC-71-C-0088.

15

32

Security Classification

DOCUMENT CONTROL DATA - R & D

1. ORIGINATING ACTIVITY (Corporate author)

Bolt Beranek and Newman Inc.
50 Moulton Street
Cambridge, Massachusetts 02138

Unclassified

3. REPORT TITLE

Natural Communication with Computers IV

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Quarterly Progress Report #15, 1 April 1974 to 30 June 1974

5. AUTHOR(S) (First name, middle initial, last name)

6. REPORT DATE

August 1974

7b. NO. OF REFS

8

8a. CONTRACT OR GRANT NO.

DAHC15-71-C-0088

b. PROJECT NO.

9a. ORIGINATOR'S REPORT NUMBER(S)

BBN Report No. 2869

9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

10. DISTRIBUTION STATEMENT

11. SUPPLEMENTARY NOTES

12. SPONSORING MILITARY ACTIVITY

13. ABSTRACT

Under this contract, Natural Communications with Computers IV broad based computer science research and development work is performed in areas including; speech understanding systems, speech compression, development of programs and programming aids, techniques for extending computer I/O capabilities, research and development on time sharing systems, and distributed computation. This research program involves the ability to represent knowledge and deal with it in computer oriented terms, requires systems capable of high degree of man-machine interaction, and draws upon many diverse fields such as linguistics, communications, programming, hardware development, speech recognition, etc. to facilitate fuller use of the enormous potential of natural modes of communication with computers.

DD FORM 1473

NONE

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
time sharing systems						
speech understanding						
distributed computation						
programming aids						
BBN-LISP						
Interlisp						
TENEX						
ARPA Network						
Computers and communication						
resource sharing						
speech compression						
vocoders						

TABLE OF CONTENTS

I. INTRODUCTION

Our accomplishments during the period April 1, 1974 and June 30, 1974 in the areas of Speech Understanding Systems, Distributed Computation and TENEX development, Languages, and Speech Compression appear in this our fifteenth quarterly progress report on Natural Communication with Computers IV.

During the preceding quarter, our Speech Understanding Project embarked on the detailed design of the new problem domain (discourse about travel budgets). With the fall 1973 Project review behind us, we have begun a new intense period of developing the Speech Understanding System and its components and interfacing these components to the new problem domain. We are expecting a new BBN TENEX machine (System D) to be available next quarter to help the heavy computational demands of the project. This new machine will be especially tuned to handle well large LISP systems (such as the Speech Understanding Programs).

The Distributed Computation project concentrated its efforts this quarter on the initial phases of access control and accounting for TIP access. We also experimented with the TELNET protocol reconnection option.

Our TENEX developments focused on the new TENEX 1.32 release to the sites which (coupled with some new TIP software) has greatly improved reliability of network connections. This release also includes core management improvements, a set of drivers for IBM-compatible tape and discs which connect to the PDP-10 via an SA10, and the capabilities for handling more than 512(10) users names on an individual TENEX system. We continued our PDP-11 based peripheral processor research and also made substantial human-engineering improvements in our mail system.

Our language research efforts continued in the LISP area. We have completed the work for compiled code overlays and released a new system to users which gives back almost 30K words of address space to the user.

During the past quarter, our Speech Compression research continued to attempt reduction of transmission rates without perceivable effect on speech quality. We have also developed some metrics for quantifying the evaluation of "speech quality".

During this quarter, the computer center put a new TENEX service system (System C) on the air with the official host-name BBN-TENEX (nickname BBN). We have changed the name of the former host which used these names to BBN-TENEXA (nickname BBNA). System C will ultimately be one of several service systems at BBN offering relatively stable, reliable

TENEX service to the ARPANET user community as well as other BBN customers. System A will ultimately be primarily a research machine with the newest software releases and for experimentation in conjunction with our research. We expect software to be tested on System A for a period before being installed on service systems.

The computer center also took delivery in late June of a Diablo style terminal (Bedford Computer Systems Inc., System 75). This report was produced on this terminal. The Diablo mechanism in conjunction with this terminal has some graphics capabilities. We took advantage of these in a new version of RUNOFF (our computerized document producer) which is now capable of superscripts, subscripts, continuous lines, and some special characters not available on the print wheel but which can be generated through the use of graphics mode.

II. CONTINUOUS SPEECH UNDERSTANDING

A. Introduction

During the past quarter, work has progressed in many aspects of the Speech Understanding system. Much effort has been spent on design of future capabilities and planning for the collection of data and the conduct of experiments in order to guide the direction of subsequent work and for discovering techniques and tuning the performance of the various components of the system. Also, we have devoted much effort during this quarter to writing papers and making the results of our research available to the scientific community.

During this quarter, we presented a collection of papers to the IEEE symposium on Speech Recognition, held at Carnegie-Mellon University April 15-19, 1974. These papers appear in the proceedings of the symposium and so far several of them have been accepted for publication in the IEEE Transactions on Acoustics, Speech, and Signal Processing. Also, a tutorial paper on Syntax and Semantics in Speech Understanding by W. Woods was presented at this symposium and is being prepared for inclusion in a volume of tutorial papers from the symposium.

In addition, we have been preparaing several papers for submission to conferences this summer. Two papers are being presented at the International Congress of Acoustics in London, "Non-determinism in Continuous Speech Understanding--Part I. Non-deterministic acoustic analysis," by Makhoul, Wolf, Schwartz, O'Shaugnessy, and Colarusso. Also, a tutorial paper on Syntax and Semantics in Speech Understanding by W. Woods was presented at this symposium and is being prepared for inclusion in a volume of tutorial papers from the symposium. "Non-determinism in Continuous Speech Understanding--Part II. Linguistic Constraints." by Woods, Bates, Nash-Webber, and Rovner. Another paper, "Linear Prediction vs. Analysis-by-Synthesis," by John Makhoul will be presented at the Speech Communication Seminar in Stockholm. Also, a paper describing the BBN Speech understanding system, "Non-deterministic Phonetic Transcription of Speech", by Richard Schwartz has been prepared for presentation at the annual meeting of the Association for Computational Linguistics in Amherst in July.

B. New Task Domain

As we mentioned in our previous progress report, work has been proceeding on the construction of a second problem domain, travel budget management. The design of this system is such that the user will not be restricted to querying the

data base, but rather he will be able to make changes to it, both hypothetical and permanent. For example:

1. What was the total cost of the trip to the Carbonell conference?
2. Bill is going to the ACL conference
3. What would be our total budget if he went to the ASIS conference instead?

(For the lunar rocks domain, while the user could both query and edit the data base of lunar sample analyses, the notion of a hypothetical change to such a data base would be very strange.)

With such a system in mind, we have been investigating the most appropriate organization and internal representation of the factual data. Using the technique of incremental simulation, we have interviewed several BBN employees who take, charge, and cancel trips. This has allowed us to observe the nature of possible user-system interactions, and has given us some insight regarding the problems that might arise.

Our simulations revealed that when asked to give information on their next trip, subjects' replies are often unstructured and ungrammatical. In this mode we observed significant speaker floundering pauses, filler words (ummm, uh), errors, and false starts. Without guidance, the speakers tend to leave out vital information. For instance,

estimated expenses are often not included. In some cases, the purpose of trip is unclear. However, in response to a small set of short questions, given below, the speakers rarely flounder and their utterances more closely resemble read speech. In addition, this type of dialogue yields a more complete description of a trip. However, there still remains the problem of possibly receiving more information than requested (e.g. Q: What is the purpose of your trip? A: To see professor X. By the way, I´ll also need a car while I´m there in order to get out to see him.) We are hoping that we can either prevent the speaker from giving more than the required information or predict the character of the extra information from the question asked.

With respect to the structure of the factual data base, we have decided to store the information in a semantic network instead of the tabular format used in our lunar rock data base. There are several reasons for doing this.

1. The factual data base can aid the speech understanding process. If the user is querying or altering data, semantics can have access to specific trips and make use of that data. For example, if it has a theory which concerns a trip and it can find a specific referent to that trip in the data base, it will have more confidence in that theory.

2. Information about specific trips will be retrieved in the same manner as information about the concept of a trip. This consistency will be useful to both the semantic and retrieval components.

3. Retrieving specific facts will be faster and more efficient because every argument to the concept of a trip (e.g. purpose, destination, data) will serve as an inverse file. This is a result of the two way links in

the semantic network. Inverse files are important because of the many different ways in which a trip can be referenced. For instance: Who has already been to California this year? Which trips did John take last month?

4. The retrieval component will require an inference mechanism. This mechanism will be easier to implement since semantic networks simplify the deduction of plausible inferences.

5. The size of the data base is small enough (the trips charged to one account number during one fiscal year) to be stored internally, rather than on disk. This makes a semantic network feasible.

6. The software for easily building and searching semantic networks already exists.

During the upcoming months we will implement this data base and construct the retrieval component. We will also construct the retrieval component. We will also finalize the data acquisition protocol.

Questions for Soliciting Trip Information

1. What is your name?
2. And the purpose of you trip is?
3. And you want to leave?
4. And you want to return?
5. You are going by means of?
6. Do you want a car? (If yes, what kind?)
7. Do you need money?
8. And the account number is?
9. Where can you be reached there?

C. Segmentation and Labeling

Early this quarter, an interactive program for specifying acoustic-phonetic experiments on retrieved occurrences of any phonetic environment was completed. It is possible to specify any subset of the data base for inclusion in the experiment, by specifying any or all of the following characteristics of an utterance: speaker, sex of the speaker, sentence, token, sampling rate, subject area,

speaking mode, and date of recording. This program permits to computation of average parameter values, values at specified points within a segment, etc., and can produce plots or tabulations of the results.

In order to facilitate statistics gathering, we've been investigating a rigorous set of rules for hand labeling sentences, since this will strongly affect end results. Some programs have been modified to make them compatible with a new speech file system which has been implemented and is much faster when only a few parameters are needed (as in experiments).

We have been holding organized, introspective spectrogram reading sessions to determine what features are more important than others in a situation in which there is only partial knowledge. These sessions have been very productive.

As a result of the Phonological Rules Workshop at System Development Corporation in June, we are setting up a mechanism for transferring and cataloguing acoustic-phonetic rules between the different sites.

D. Lexical Retrieval and Word Verification

1. New Phonetic Dictionary

The phonetic dictionary has been re-designed to include information about word pronunciation that is useful both for lexical retrieval and word verification. Additions include syllable boundary markers, glottal stops, more complete information about missing and extra segment probabilities, stress for consonants, and relative likelihood measures for alternate phonetic spelling fragments. The set of "phonemes" was expanded to include the phonetic elements with which the synthesis programs deal. A representation for the phonetic dictionary which can be used by both the lexical retrieval programs (which deal with a phonemic transcription) and the synthesis programs (which deal with phonetic elements) was designed. The phonetic dictionary was formulated and keyed in, and programs for reading it were written and debugged.

2. Interface to Word Verification

During this quarter, we have worked towards implementing an initial version of the word verification component. Much of the effort has involved integrating the subcomponents with one another and with the total speech understanding system. A control strategy for obtaining word hypotheses and coordinating the synthesis and mapping

activities has been specified and partially implemented. This allows the verification component to interface directly to the word proposer and will give us the opportunity to examine different scoring strategies for specific segment types. Initially, the word verification component will be studied as a way to refine the word-match quality evaluation done by the lexical retrieval programs.

E. Syntax

During this quarter, work has been devoted to expanding and improving the grammar and shaking out and fixing bugs in the parser. We have written and checked out a grammar for spoken numbers dealing with all of the alternative forms of numbers that we expect to encounter in the travel budget domain, and we are beginning to develop a similar grammar for dates. Other grammar extensions are planned, and the development of the parsing algorithm for providing useful interaction between syntax and semantics and other components is continuing.

F. Semantics

In building a semantic network for the new travel budget management domain, we decided to attack the problem of characterizing the objects in the domain so as to capture the likely ways that those objects would be discussed. This would be in addition to characterizing the relations in the

domain (primarily, verbs and nominals), which we found so useful in the lunar rocks domain. For example, as well as indicating that "spend" relates instantiations of the person concept, the money concept, and the concept of things which, unlike the best things in life, are not free; we want to represent that a budget contains entries which may each detail. among other things, an expense, an amount of money, either estimated or actual, and the person responsible for the expense. And this in a way that will allow us to say that the first four sentences shown below are plausible utterances in this domain while the last is not.

> How much money is left in this year's budget?
> What trips have been taken that were not in the original budget?
> If we take three trips to LA next month, will we still be within budget?
> Give me a breakdown of the travel budget.
> Tell me about John's first budget for overhead.

In the semantic network designed for the lunar rocks domain, we finessed the problem of characterizing objects, by representing in the network and in case frames as many of the local syntactic environments in which discussions of the objects could be constructed. For example, rather than describing a rock as composed of fused together chunks of minerals (which themselves are structured arrangements of elements), and having a "rule" that "if X can be a part of Y, one can talk about Y's with X, Y's which do not have X, etc.", we had a node representing the construction "with a constituent" (i.e. an element or a mineral) which could

restrict any instantiation of a set of lunar rocks. While such a surfacy description was somewhat useful for low-level predictions based on the appearance of related words in the word lattice, it gave no handle on such higher-level predictions as "the user's going to talk about the composition of lunar breccias". Therefore the above constructs are likely.

We are currently investigating using frame-like structures to characterize the objects in the domain so as to imply how to talk about them. (These structures would be nodes in the semantic network, so as not to lose its many advantages.) For each object, a frame would contain information about the properties of that object: what they were, how many values each could simultaneously have for the same object, how closely each was related to the object, etc. Each of these pieces of information would be useful in making predictions about how a discourse about that object would run. For example, the frame for "trip" would specify that one property of a trip was the person taking that trip, that only one person could be associated with a trip, and that this property was very strongly associated with trip. Then, we could predict very strongly, given a match for "trip" in the word lattice, that there may also be an instantiation of the person concept. Having found one, we would also know not to predict another within the same theory.

In the next quarter, we will continue with the construction of the semantic network for the travel budget management domain and the functions for using the above mentioned object descriptions for prediction. With the lexical retrieval fork for the domain already built, we will also be able to test out its predictive ability.

5. User and Task Model

During this past quarter, we began the task of formulating a task model for the travel budget management domain. This is an attempt to generalize the behavior of an arbitrary, goal-directed user of a system such as this one which permits both the querying and the real and hypothetical alteration of its data.

Based on simulated dialogues with the travel system we envisage, we have characterized several different possible modes of interaction with the system and transitions between them. A session with the system then consists of a sequence of interaction modes, which are themselves built out of other modes and intents. An intent is the smallest unit in our task model and represents the supposed purpose behind an utterance made by the user. An intent is, of course, somewhat sensitive to the mode one has hypothesized for the user. For example, if the user were to say, in edit mode, "Craig is also going to the ACL Meeting.", one would say his intent was to make a permanent change to the data base. In

query mode, however, (with perhaps a change in the intonation), one would say it was to get information from the data base.

Currently, we have characterized the following modes of interaction:

a. add - the user is attempting to add new information to the data base.

b. conflict - the system has pointed out a contradiction between some statement or assumption made by the user and its own information. The user must then respond to it. (That he will respond in some suitable way is one of the pragmatic assumptions of our system.

c. edit - the user is attempting to change some information already in the data base.

d. q-c - the system does not understand either partially or completely, the user's utterance and asks for clarification. It is again a pragmatic assumption that the user will respond in a proper manner.

e. query - the user is attempting to get information from the system.

f. supposition - the user is making hypothetical changes to the data base to see where they will lead.

g. test - the user is attempting to ascertain that the system's knowledge about some past or future event conforms with his own.

While it would be too much to detail here the structure of each of the above modes, we hope it would be interesting to describe one. A user enters _edit_ mode with the intention of changing some information in the data base. As a result of his utterance,

a. the system may ask for clarification. That is, the mode may switch to _q-c_. Upon successful clarification, things proceed as in c. below.

b. the system may point out a contradiction. For example, the user may have a mistaken assumption about what is actually in the data base. Here the mode switches to _conflict_.

c. the sytem may make the requested change and confirm to the user that it has made it. At this point, the user may want to make another change, remaining in _edit_ mode, or leave that mode for another one.

Work will continue in the next quarter on further characterizing the modes of interactions, the types of utterances within each, and those types signalling transitions between them.

III. DISTRIBUTED COMPUTATION AND TENEX IMPROVEMENTS

A. Introduction

This quarter, the security aspects of the RSEXEC system were greatly enhanced by use of the fork group facility recently added to the TENEX operating system. Each instance of service, (i.e. each user requesting access to the distributed file system) is created in a separate (TENEX enforced) protection domain that enables him to reference only the files at each site he would be able to reference if he had gone through a manual login to that site. This change resulted in considerable simplification and speedup of the RSEXEC system, so a similar change is now being considered for the FTP (file transfer) server.

We have completed the initial stages of an RSEXEC system for providing access control and accounting for TIP access, and expect to provide an experimental version of this system during the next quarter. With this system, each TIP user will be initially (automatically) connected to an instance of RSEXEC and required to provide a network login name and password for authentication and accounting. Once he has "logged into the network" he may use all the services of the RSEXEC system, and finally log into some specific service site to do programming work.

The TENEX 1.32 release included the network reliability improvements described in RFC #636. TIP users (with TIP version 322 and greater) will see host service interruptions reported on their consoles, followed by either a resume or restart message when host service is restored. Typically, the user will be able to continue use of his connection as if there were no interruption of service.

B. Meetings

On May 1, a Packet Radio meeting was held at BBN with representatives of Collins Radio and Stanford Research Institute to explore the issues of the Packet Radio Station design. Two documents resulted from this meeting: a preliminary design specification for the functional components of the station, and a specification for the standard packet interface to interconnect the station PDP-11 and the digital packet transceiver developed by Collins. These documents were published as Packet Radio Temporary Notes #104 and #105.

On May 6, we hosted a panel session entitled "Applications and Extensions of the TENEX Operating System" at the National Computer Conference in Chicago. Panelists included representatives of the ILLIAC-IV Project of NASA Ames, Xerox Palo Alto Research Center, the University of California at Santa Barbara, Digital Equipment Corporation, and BBN. The session was well received and drew many

questions from several hundred attendees.

On May 28 and 29, we attended a Packet Radio meeting hosted by Collins Radio in Washington, D.C. to discuss Packet Radio system design issues. There was spirited discussion of several controversial design issues, with agreement to hold ongoing exchanges to resolve these issues.

On June 12 and 13 we attended a meeting of the newly-formed TENEX Advisory Committee at ISI in California. Members represented ISI, SRI, CCA, BBN and ARC. A wide ranging discussion helped to generate a report covering development procedures and desired features for future TENEX releases. A tentative schedule for near-future developments was agreed upon, and is being implemented. In the advisory capacity, the committee compiled a list of areas where it felt able to provide ARPA with some policy suggestions. A list of questions was referred o ARPA/IPTO for consideration.

A series of meetings with the BBN-IMP group during this quarter resulted in a minor revision of the host-IMP interface behavior to simplify construction and programming of the host interface. The changes to BBN Report #1822 are documented in a memo distributed to the Technical Liaisons in July. The new definition of the standard host interface Ready Line Control was provided in RFC #642. These specifications have been included in the design of the SDC

HSI-11B, the ARPA standard PDP-11 local host interface.

C. Distributed Computation

1 RSEXEC Developments

RSEXEC has been modified to handle so-called TENEX "files-only" directories.* The main problem to be solved in doing so was that of properly controlling access to remote files-only directories. Insuring proper access control was accomplished by introducing the concept of "primary" directories. For each site in his RSEXEC profile, a user specifies (implicitly or explicitly) a primary directory which must be a login directory at the site. The user's access to files-only directories at a given remote site can then be based upon the access granted to his primary directory at that site.

The TIP-RSEXEC and RSSER (the RSEXEC server program) have been modified to make use of the new fork group and terminal PSI features of TENEX (see BBN Report 2822). The TIPSER program has been changed to establish a fork group for each instance of TIP-RSEXEC service. This permits each

---

* A files-only directory is one for which logins are not allowed and therefore which can be used only to catalog files. At the option of the responsible user, access to a files-only directory and the files contained in it can be controlled either via a directory password or the TENEX group access mechanism.

service instance to run within its own independent access control environment and to set up a terminal interrupt structure (e.g., control-C,control-T) for its own fork controlling terminal. With this change, TIP-RSEXEC need no longer attempt to simulate terminal interrupts. This has resulted in improved terminal interrupt behavior and the potential for providing within TIP-RSEXEC services requiring sophisticated terminal PSI capabilities (TIP-RSEXEC was unable to faithfully simulate the TENEX terminal PSI system).

The RSSER program now creates a fork group for each service instance. When a remote user process identifies itself (via name and password) RSSER uses the fork group "proxy login" capability to establish a proper access control environment for serving the remote process. RSSER need no longer simulate TENEX file system access control in order to guard against compromising the privacy of user's files. This change to RSSER has had several beneficial effects. Because RSSER need no longer run as a privileged system job, it has had the effect of reducing the TENEX "security kernel" (by removing RSSER from it). Furthermore, it has allowed a significant simplification to the RSSER program. Finally, since it need not simulate access control, RSSER is much more efficient. We have observed a factor of five (5) speed up in the RSEXEC directory acquire function as a result of this change.

We will be able to distribute the improved TIP-RSEXEC and RSSER programs to ARPANET TENEX sites after TENEX version 1.32 is installed at the various sites.

During this quarter the RSEXEC TELNET facility was augmented to support the "new" TELNET protocol. RSEXEC currently supports both "old" and "new" TELNET protocol. The old protocol is normally used unless the remote site initiates new protocol interactions or the user explicitly requests new protocol.

A SCHEDULES command has been added to TIP-RSEXEC which enables users to print scheduled down times for IMPs, TIPs and network service sites. The schedules data base is maintained by the Network Control Center at BBN.

The RSEXEC system was installed on the PARC-MAXC, ISI-DEVTENEX and BBN "System C" hosts during this quarter. This brings the number of sites which regularly run the RSEXEC server program to 11.

2. Experiments with Reconnection Protocol

A prototype implementation of the TELNET protocol reconnection option was completed this quarter. The reconnection protocol provides a means whereby one process (B) can reconfigure a communication path between itself and another process (A) to be between the second process (A) and a third process (C). (For a detailed discussion of

reconnection see RFC #369.) Using reconnection a serving process (B) can switch a using process (A) off to another serving process (C). For example, the TIP-RSEXEC (process B) could use it to hand a TIP user (process A) off to a service site (process C) after the TIP user has properly authenticated himself, read the latest net news, obtained host status information and finally indicated his desire to use the service site.

The prototype implementation was done within the context of RSEXEC. Process A was a TIP-RSEXEC, process C a modified RSSER (to simulate a server TELNET) and process B a modified RSEXEC (to simulate a TIP). Our experimentation has revealed a few inadequacies in the protocol as specified. In addition, it has served to strengthen our conviction that reconnection is a basic function that should be supported at the Host-Host protocol level. A document is in preparation detailing our experience, the protocol inadequacies together with our corrections to them, and some recommendations for other implementers.

3. TIP User Authentication and Accounting

Together with the BBN-TIP group we have initiated a joint project to provide TIP user authentication and TIP usage accounting. The first step in the project was to prepare and to submit to the ARPA office a project plan detailing how TIP authentication and accounting will be

accomplished. The plan was approved by ARPA and implementation of the system is currently underway.

User authentication for TIPs will be achieved via the TIP-RSEXEC system as previously suggested (BBN Report 2721). TIP usage accounting will be accomplished in a similar manner: each TIP will send accounting data to an RSEXEC accounting server periodically and whenever a user terminates a TIP session; this "raw" accounting data will be regularly reduced to produce usage accounting summaries.

Due to the high priority given the TIP authentication and accounting project by the ARPA office, work on the Coupled Message Service has been temporarily suspended.

D. <u>TENEX Improvements</u>

1. TENEX 1.32 Release

TENEX version 1.32 was released at the end of this quarter and is being transported to other TENEX sites. Most of the features of this release have been previously reported. Additional features in the release are described below.

TENEX 1.32 contains a complete implementation of the protocol augmentation described in NWG RFC #636. The reliability of network connections has been greatly improved as a result. It is now possible for the TENEX monitor to

stop at a breakpoint and be continued a few minutes later without any disruption of network terminal connections beyond that experienced by local terminals. In fact, the disruption has less impact because the user is informed explicitly that there is a disruption rather than having to infer the existance of the disruption from the lack of response. TIP users (with TIP version 322 and greater) benefit from these improvements as well as users using TELNET from a TENEX running version 1.32.

Another feature which was installed in TENEX 1.32 during this quarter permits the network host information to be obtained from a file instead of from assembled tables. It was previously necessary to patch these tables whenever a new host was added to, deleted from, or moved within the network. This is now accomplished by editing a text file.

Another feature in 1.32 which has not been previously reported is an augmentation of the SPACS JSYS to permit users to lock down pages of memory. This permits certain real-time programs to run which otherwise would fail due to inopportune page faults. The implementation is such that the user cannot accidently abandon a locked page or confuse the system by locking a page twice. Access to this feature is permitted only to certain privileged users.

TENEX 1.32 has had major changes to the user index and file directory structure to permit more than 512(10) user names on each TENEX system. The current limit is 16,000(10) and is an assembly parameter. Some subsystems have lower limits (e.g. ACCT10 currently has a 1000(10) limit due to table size limitations).

2. Pie-Slice Scheduler

The pie-slice scheduler, currently under development, is intended to satisfy the needs of installations requiring the capability to guarantee groups of users varying minimum levels of service.

A user logging on to TENEX is assigned to a pie-slice group as a function of the account designation. Each pie-slice group has associated with it a certain fixed share of the available non-overhead processor cycles. The pie-slice scheduler will guarantee that when a pie-slice group is represented by one or more processes actively requesting service, the total time devoted to those processes will not be less than the group's fixed share. In an attempt to equalize the cost effectiveness observed by each group over the fiscal period, a portion of the shares belonging to unrepresented groups (groups for which there are no jobs) is assigned to the currently least-cost-effective group. This portion will be an operator-settable parameter.

The pie-slice scheduler will be distributed to TENEX sites as an option selectable at system-assembly time. The work is currently in the debugging stage.

3. Core Manager Improvements

During this quarter, various detailed improvements have been made to the balance set management policy to provide better response to interactive processes while guarding against over-commitment of main memory.

The routines for post purging of working sets have been revised so as to be compatible with the current core manager. Preliminary experiments were performed to determine whether use of post purging would reduce the processor cost of memory garbage collection. The results of these experiments were inconclusive and more extensive testing is planned subsequent to completion of the pie-slice scheduler.

4. SA-10 Driver

Three ARPANET sites are installing, or have installed, a Systems Concepts, Inc., SA10 subsystem adaptor to control IBM-compatible disk and/or tape systems. These are BBN, CCA (Computer Corporation of America), and ISI, (University of Southern Carlifornia, Information Sciences Institute). During this quarter, we received an SA10, a Calcomp

3330-equivalent disk system, and a Storage Technology tape system.

As a result of the problems encountered during installation of the disc system, BBN rewrote significant portions of the diagnostic routine provided by Systems Concepts, incorporating features requested by Calcomp. Two important results of this effort were: 1) a much improved diagnostic was provided to CCA (which already had a Calcomp disk system), and to ISI (which was about to install one), and 2) the working relationship with Calcomp was improved greatly, which also benefited the ISI installation.

The integration of new software to drive these I/O devices into TENEX was nearly completed during this quarter, in preparation for BBN's new service host. The resulting code will be distributed, with TENEX version 1.32, to ISI for their use.

5. TENEX Security Study

At the request of the ARPA office, a study of TENEX performance in the area of operating system security has been initiated. As part of this study, we produced and submitted to the ARPA office a paper called "A look at TENEX Security." The paper states TENEX security goals, summarizes the TENEX mechanisms for access control and protection, assesses how well TENEX meets the stated goals

and makes a number of recommendations for improving TENEX security. In addition, the paper catalogs known TENEX security problems.

Several of the recommendations have already been carried out including preparation of a User Security Manual which we intend to distribute to all TENEX users. We are continuing to work toward improving TENEX performance in the security area.

E. Peripheral Processor

1. Packet Radio

We have selected a PDP-11/40 for the packet radio station. It will support packet radio application programs running as user processes under the ELF operating system developed at Speech Communications Research Labs. All applications modules will be coded in BCPL to maximize machine independence and obtain the obvious advantages of program development in a high-level language.

The interface between the PDP-11 station and its associated packet radio transceiver has been specified: it is called the Standard Packet Interface. This interface provides full duplex 16-bit parallel data transfers in both directions, along with asynchronous 4-way handshake control signals, packet delimiting signals, and reset signals. The

PDP-11 interface will be implemented using two standard Digital Equipment Corporation DR-11B memory channel interfaces attached to the Unibus.

2. Cross-net Debugging Protocol

A protocol was designed for debugging and bootstrapping PDP-11's over the ARPANET. The protocol is described in detail in forthcoming RFC #643. The design of the protocol is such that it can be used to debug processes running under the ELF operating system and to debug processes running "stand-alone" on the PDP-11.

The protocol was used to implement a network bootstrap loader which can be used to load a PDP-11 from TENEX via the ARPANET. The PDP-11 part of the network bootstrap program is less than 400 (octal) bytes long, and can be loaded into the PDP-11 by the ROM bootstrap loader from paper tape. The PDP-10 part of the bootstrap loader can load either binary files produced by DEC assemblers (loadable by the absolute loader) or .SAV files produced from output of the PAL11X assembler.

3. Standard PDP-11 IMP Interface

We served to chair an ARPA-appointed committee charged with specifying a standard PDP-11 host interface which could be obtained and maintained as a standard vendor product,

possibly by Digital Equipment Corporation as a standard PDP-11 peripheral product.

The committee reviewed interfaces designed by the ANTS project at the University of Illinois, by System Development Corporation, by the Information Sciences Institute of the University of Southern California, and by the ILLIAC-IV Project at NASA Ames. The SDC interface was selected as the least costly to produce and maintain as it is constructed entirely of DEC standard components. However, the initial version of this interface, the SDC HSI-11A, had a number of deficiencies in its control of the host and IMP ready lines. We published a complete specification of the correct operation of these lines in RFC #642, "Ready Line Philosophy and Implementation". Subsequently, the HSI-11A was redesigned to conform to this specification, and designated the HSI-11B. Once the prototype unit has been demonstrated, vendors will be asked to quote on production and maintenance of these standard interfaces.

F. Mail System Improvements

1. Mail Sending

The handling of network mail has been greatly improved. A study of the error codes sent by the receiving host and the interpretation of these codes by the sending host revealed that the FTP error codes as currently defined were

inadequate for the intelligent handling of mail. In particular, the codes did not distinguish between temporary (e.g. mailbox busy) and permanent (e.g. no such user) failures. We wrote an RFC #630 which suggested new standards for using the existing error codes so as to introduce that distinction. The TENEX FTP server was modified to send these new error codes (along with informative messages), and MAILER and SNDMSG were modified to make use of these codes in order to decide whether to declare the mail undeliverable or keep trying. Previously MAILER and SNDMSG considered all failures fatal, since they couldn't interpret them. These program changes, while improving mail transfer among TENEX sites, do not introduce any incompatibilities with non-TENEX sites. In addition to improving mail transmission in the short term, our study provided valuable input to the design of new FTP replies.

A number of additional improvements have been made to both SNDMSG and MAILER. Error handling and error messages have been improved both for local and network mail. Address list input in SNDMSG has been improved by: accepting octal host numbers, in case trouble is encountered with the host name; allowing groups to be delimited, so that both grouped and ungrouped names may be entered; separating addresses into "to" and "cc"; allowing messages to be addressed to arbitrary (local) files as well as to users. It is now possible to invoke TECO from within SNDMSG to edit the text

of the message being composed. Duplicate local addresses are eliminated in SNDMSG, so the use of overlapping distribution lists (of local names) does not cause multiple delivery. Commands were added to SNDMSG to force queueing or modify the time parameters used in sending. MAILER has been changed to prevent unauthorized sending of login messages, and has been made more efficient in local mail delivery.

2. Mail Reading

In preparation for the forthcoming new mail system (see section 4 below) a slight change to the format of message files was specified. RD and READMAIL were modified to handle this new format (in addition to the old) so that the transition between systems will be smooth.

RD's efficiency was greatly increased by modifying it to use the new TECO described in III.G.2.

3. Mail Forwarding Service

The addition of a second TENEX host at BBN has necessitated new capabilities for handling computer mail. In particular, mail arriving at either system must be delivered to the addressee, regardless of the system his mailbox is actually on. Otherwise, both local and remote users sending mail to users at BBN would have to remember

which of the BBN TENEX hosts each addressee uses to maintain his mailbox. This could be confusing since many people maintain directories on both systems, but a mailbox on only one.

To avoid this confusion we have implemented local mail forwarding as an auxiliary system (subsys) program called the "mailbox finder." The mailbox finder may be run by system mail handling routines (FTP, MAILER, SNDMSG), or may be run as a user program. The current version accomplishes redirecting of mail from synonymous names to the correct mailbox name. In one case, a programmer has two disk (login) directories; mail addressed to his alternate name is automatically redirected to the mailbox under his primary name. In another case, a husband and wife agreed that all of her mail, most of which is mis-addressed (by sender) urgent messages to him, be rerouted to him. As a user program, it is useful to a person wishing to identify the correct address for mail they wish to send, or even to ascertain whether a particular user maintains a mailbox at BBN at all.

We have uncovered several major issues related to more general automatic mail forwarding. These include addressing mail by human name (possibly with spelling correction or assistance) rather than login name; naming person and/or computer site not specifically but by class (defined either geographically or politically); forwarding mail to foreign

sites, or else returning a special negative acknowledgement such as, "He's not here, but I have a Jones at OFFICE-1 and one at ISI;" authenticating mail which has been forwarded through various sites; and automatic distribution of mail to members of a group. The current version incorporates a mechanism for the last of these, but distribution facilities currently being added to FTP are needed before this feature will be functional.

4. New Mail Reading System

A new mail handling facility has been designed which is intended to replace the SNDMSG, RD, and READMAIL programs on TENEX. The new Mail program utilizes a command language based on the TENEX Executive language to read and generate messages.

As one of the initial steps in the design of this system, a users' manual which specifies exactly the user interface to the system has been prepared and distributed to the ARPANET user community (via he USING Group) for review and comments. This was done to allow the users of the system to exert some influence on the system design.

Although user review of the design is not yet complete, preliminary user comments indicate that no major design revisions are required. On that basis, coding and debugging have started. Implementation of the mail system has been

split into two phases, both to simplify the implementation and to provide a useful program as soon as possible. The first phase consists of the mail reading portion of the system. Coding for this portion is complete, and final debugging is under way. The second phase of the implementation is to provide all of the facilities for the generation and dispatching of messages. Most of the coding for the second phase is complete and debugging will start with the completion of phase one.

G. Other Subsystems

1. EXEC

Several security-related changes have been made to the EXEC. Users are now notified at both LOGIN and LOGOUT the times of other jobs logged in under the user's name. At LOGIN the user is informed of the date and time of his most recent prior use of the system.

Users may now change the password of any directory for which they know the password.

To assist users in controlling the protection of their files a more "English-like" command has been added:

ACCESS (TO FILES) <file list> (BY) <access path> (IS) <access>

<access path> is one or more of the following words separated
by commas: SELF, GROUP, OTHERS, ALL.

<access> is one or more of: READ, WRITE, APPEND, EXECUTE, PAGE-TABLE, NORMAL, ALL, NONE.

The commands PERPETUAL <file list> and NOT PERPETUAL <file list> have been implemented. Perpetual files cannot be deleted by normal means and are protected from the backup system.

Selective additions and changes provided by other sites on the ARPA Network have been included.

2 TECO

TECO now inputs files an order of magnitude faster by using the PMAP system call in order to map pages directly from the file. This was made possible by the elimination of the use of EOL (37) in many areas of TENEX. In particular, TECO used to have to convert carriage-return-linefeed sequence to EOL on reading in a file which previously required a character at a time read in approach.

IV. LISP

During this quarter we have completed the overlays for compiled code and have released a new system to users. The new system uses 27,000 fewer words of address space than the previous system while including more user facilities.

We have completed one more step in the multiple environments; that of merging with overlays. A test system has been released to selected users. The block compiler is the last remaining task prior to release of the multiple environments.

As the first step toward measuring the memory requirements (working set) of INTERLISP and obtaining timing breakdowns, we have cleaned up and modified an existing PDP-10/TENEX simulator. The simulator will permit us to make detailed studies of the memory reference patterns of a variety of INTERLISP tasks and to compute accurate timing information.

New features added to INTERLISP include the ability to read from strings and an extension to the concept of syntax class in read - macros. We have also added user-interrupts. That is, a user program can assign terminal interrupt characters and handle the interrupts in a completely general way. An interrupt character can be defined either as a "hard" interrupt, that is, to occur immediately; or a "soft" interrupt to occur at the next function call.

We visited Dr. Alan Bond of Queen Mary College, University of London, to discuss the problems involved in implementing INTERLISP. The Artificial Intelligence group at Queen Mary College is working with the government computing center to implement an INTERLISP on an ICL. They have chosen INTERLISP because of its user orientation and the size of the existing user community. A major goal is easy communication of programs and ideas between themselves and the current community of INTERLISP users.

V. SPEECH COMPRESSION

The major effort in our speech compression research for the past quarter has been in developing encoding schemes which would further cut down the transmission rate without any perceivable effect on speech quality. Two of the encoding schemes that have proved quite successful are: (a) variable wordlength encoding scheme, and (b) variable rate encoding scheme. We found that with the use of these encoding schemes, good quality 10 kHz sampled speech can be obtained at transmission rates as low as 1650 bps. Syntheses obtained using these encoding schemes were demonstrated at the May meeting of the ARPA Network Speech Compression (NSC) group for a rather difficult data base involving a dialogue between a male and a female speaker. We have also made preliminary investigations into the objective evaluation of speech quality.

In our speech compression project, we have worked on two types of encoding schemes for transmission parameters. The first of these, known as variable wordlength encoding, takes advantage of the probability distributions of the transmission parameters and encodes each of them using a variable number of bits. The second scheme is called variable rate encoding; it transmits the paramaters only when the speech characteristics have sufficiently changed. In our low bit-rate linear predictive vocoder that uses 10 kHz sampled speech, the variable wordlength encoding offers

an average saving of about 600 bps while the variable rate encoding cuts the average transmission rate by about 450 bps. With the use of these two encoding schemes, the transmission rate drops to as low as 650 bps. Also in the last quarter, we have developed two potential objective measures for the evaluation of speech quality.

Three NSC notes have been written in the past quarter [1-3]. Two additional NSC notes are now being completed [4,5].

1. Variable Wordlength Encoding

We have been investigating various information theoretic techniques for coding the speech parameters for transmission. We found that two techniques, Huffman coding and delta encoding, are particularly useful in reducing the transmission rate, or equivalently, improving the quality for a fixed transmission rate. Reductions of approximately 20% in the transmission rate have been common. These techniques use the statistics of the speech parameters to determine the particular values that are most likely to be transmitted, and then code these values with fewer bits. The number of bits, or wordlength, required for a particular set of parameter values is variable. Neither of these techniques results in information loss, but only in more efficient transmission of the information

(a) Huffman Coding

Huffman coding, as described in [7] and [8], offers several advantages. First, it is essentially independent of the acoustic model chosen. It is approximately equally efficient for coding reflection coefficients, log area ratios, or variable rate transmission coefficients. Second, it does not require the parameters to be quantized such that the number of quantization levels is an integer power of 2. For example, Huffman coding results in efficient transmission for a parameter that has 17 quantization levels. Straight binary value coding would require five bits for this parameter, with most of the fifth bit being wasted. With Huffman coding, the quantization of a parameter can be chosen to conform to other criteria, such as equal quantization step size, or equal spectral error. Finally, Huffman coding has been proven optimal. That is, the average transmission rate is the minimum possible. For the particular type of Huffman coding we are using, the maximum length of the parameter code is also minimized. This latter property allows reasonable limits on the word length to be found. Because Huffman coding is optimal, it also provides a useful standard for comparing other encoding methods.

Huffman coding has certain drawbacks, however. It requires the use of tree structures for decoding, resulting in increased storage requirements. It may be possible to combine the trees for a number of parameters, thus reducing the storage required. It also introduces additional complexity into the packing and packetizing algorithms, because of the variable wordlength. For these reasons, it will not be implemented for the December network demonstration, but we hope to implement it shortly thereafter.

(b) Delta Encoding

We have also investigated coding the change in a parameter from frame to frame. For some parameters, notably pitch, which change slowly but which require a large number of quantization levels, this seems to be a good technique. We assume that a change of zero is the most likely change, and code this with one bit. Then the other changes are coded with one more than the usual number of bits.

Using Huffman coding after delta encoding is also useful. The delta encoding removes some of the speaker dependent aspects of the parameters. For example, the change in pitch for a female speaker is likely to be nearer that of a male speaker than are the actual values of pitch. The delta encoding thus improves the statistics for Huffman coding, and also reduces the chances of an anomalous

speaker.

The delta encoding technique adds very little in terms of complexity to the coding algorithm, but it does present an appreciable cost to the packing and packetizing algorithms, again because of the variable wordlength.

2. Variable Rate Encoding

A variable rate or a dynamic encoding scheme transmits parameters only when the speech characteristics have sufficiently changed. Parameter transmissions occur more frequently when speech characteristics are changing rapidly as in phoneme transitions, while the transmissions are spaced further apart when speech characteristics are relatively constant as in steady state sounds. As compared to a constant rate transmission system, the variable rate transmission system could, if designed properly, yield lower transmission rates at better speech quality in transitions and without any perceivable effect in steady state regions.

To determine if speech characteristics have sufficiently changed since the last transmission, we have used a measure that is the logarithm of the ratio of the mean-squared values of the error signal (residual) obtained when (i) the optimal linear predictor parameters are used and when (ii) the last transmitted parameters are used. If the predictor parameters are assumed to have Gaussian

probability distributions, then this measure is the same as the log likelihood ratio [6]. To see how our encoding scheme works, let us suppose that we have decided to transmit the parameters for frame 1. For frame 2, the optimal linear predictor parameters are determined along with the minimum mean-squared value of the error signal. Using the predictor parameters of frame 1, the mean-squared value of the error signal is also determined for the speech signal of frame 2. The logarithm of the ratio of the two mean-squared values is compared against a threshold. If the threshold is not exceeded (success), the data for frame 2 is not transmitted; however, data transmission occurs if the threshold is exceeded (failure). In the former case, the same procedure is repeated for the successive frames until a failure occurs or the number of consecutive successes exceeds a preset limit. When one of these two conditions is satisfied, the data for frame 1 is transmitted along with the number of consecutive successes. At the receiver, we interpolate between parameter receptions to generate data at a rate equal to or greater than the rate at which parameters are extracted at the transmitter.

In our speech compression system provided with the variable rate encoding, we have used an analysis rate of 100 frames/sec (i.e., parameters are extracted once in every 10 msec). A satisfactory value of the threshold for the log measure was found experimentally as 1.5 dB. Parameter

transmissions were not allowed to be spaced by more than 80 msec (8 frames). Dynamic encoding was done only on log area ratios. Pitch and gain were transmitted, at a constant rate of 50 times/sec. With these specifications, we experimented with 14 sentences of speech material from 10 speakers (male and female). The frame rate of transmission for log area ratios varied between 24 and 45 frames/sec, with an average of 37. The transmission bit-rate varied between 1800 and 2600 bps, with an average of 2200. In comparison, a constant-rate transmission system operating at 50 frames/sec yields a transmission bit-rate of 2650 bps. Thus, the variable rate encoding offers an average saving of 450 bps. Informal listening tests gave a slight edge to the dynamic encoding scheme over the 50 frames/sec constant rate scheme.

Currently, we are working on other methods for detecting when sufficient changes in speech characteristics occur.

3. Measures for Objective Evaluation of Speech Quality

As explained in our last QPR, one of our goals in developing measures for objective evaluation of speech quality is to be able to make relative judgments of small differences in speech quality which are difficult to detect through informal listening. In the last quarter, we have formulated two candidate measures for this purpose. The first one is the spectral error between the synthesized

speech and the original. At equally spaced time instances, the linear prediction spectra for the original and the synthesized speech are computed, and the absolute error between the two log spectra averaged over the entire frequency range is found at each of these time instances. For the second measure, log area ratios are computed for both the original and the synthesized speech, and the averaged absolute error between these log area ratios is evaluated at the various time instances. The time history of the spectral or the log area ratio error within a speech utterance, the time-averaged value of the error and its variance, and the maximum error will all be used in the objective evaluation of speech quality. Specifically, we found that the error (spectral or log area ratio) due to interpolation is much larger than the error due to quantization. This has reinforced our belief that better interpolation schemes (rather than simple linear schemes) should be developed to yield better quality speech. We plan to investigate the usefulness of the two measures mentioned above for the objective evaluation of speech quality.

4. Real Time System Implementation

We have been involved in several aspects of the real time implementation effort. We have configured the SPS-41/PDP-11 system, and are expecting delivery of it during the next quarter. In preparation, we have completed

a programming course from SPS, Inc.

We have participated in the design of the ELF operating system for the PDP-11, and will continue to do so as much as possible. In cooperation with other sites, we have designed and are implementing some support software for the SPS-41. This consists of an automatic reformatting package that will assist in the preparation of large SPS programs made up of many smaller overlay segments. The preparation and ordering of these segments is at present a time consuming, tedious task, one that is prone to error and very difficult to debug. The reformatter will reduce these problems greatly. We are also consulting with other sites in the preparation of SPS programs. This cooperation has been of considerable mutual value in the past, and we are sure that it will continue to be so.

REFERENCES

1. J. Makhoul, "Selective Linear Prediction and Analysis-by-Synthesis in Speech Analysis," NSC Note #16, May 1974 (Also BBN Report #2578).
2. J. Makhoul and R. Viswanathan, "Quantization Properties of Transmission Parameters in Linear Predictive Systems," NSC Note #17, May 1974 (Also BBN Report #2800).
3. J. Makhoul, R. Viswanathan, L. Cosell and W. Russell, "BBN Working Papers on Speech Compression - I," NSC Note #24, May 1974.
4. R. Viswanathan and W. Russell, "Quantization Routines for Linear Predictive Vocoders," NSC Note #33, July 1974.
5. L. Cosell and J. Makhoul, "Variable Wordlength Encoding," NSC Note #34, July 1974.
6. F. Itakura, "Minimum Prediction Residual Principle Applied to Speech Recognition," Proc. IEEE Symposium on Speech Recognition, CMU, Pittsburg, PA., 181-185, April 1974.
7. D.A. Huffman, "A Method for the Construction of Minimum-Redundancy Codes," Proceedings of the I.R.E., Vol. 40, 1098-1101, September 1952.
8. E.S. Schwartz, "An Optimum Encoding with Minimum Longest Code and Total Number of Digits," Information and Control, Vol. 7, 37-44, 1964.